Fast Semi-supervised Evidential Clustering

Violaine Antoine[a,*], Jose A. Guerrero[a], Jiarui Xie[a]

[a]*Clermont Auvergne University, UMR 6158 CNRS, LIMOS, F-63000, Clermont-Ferrand, France*

Abstract

Semi-supervised clustering is a constrained clustering technique that organizes a collection of unlabeled data into homogeneous subgroups with the help of domain knowledge expressed as constraints. These methods are, most of the time, variants of the popular *k-means* clustering algorithm. As such, they are based on a criterion to minimize. Amongst existing semi-supervised clusterings, Semi-supervised Evidential Clustering (SECM) deals with the problem of uncertain/imprecise labels and creates a credal partition. In this work, a new heuristic algorithm, called SECM-h, is presented. The proposed algorithm relaxes the constraints of SECM in such a way that the optimization problem is solved using the Lagrangian method. Experimental results show that the proposed algorithm largely improves execution time while accuracy is maintained.

Keywords: Evidential clustering, label constraints, constrained clustering, theory of belief functions, optimization

1. Introduction

Clustering is a knowledge discovery approach that aims at grouping objects according to a notion of similarity based on the characteristics of the objects. Clustering algorithms are divided into two families: hierarchical clustering that builds a hierarchy of clusters and partitional clustering that generates disjoint subsets of the data. Partitional clustering methods include *k-means* and its

*Corresponding author
Email address: violaine.antoine@uca.fr (Violaine Antoine)

variants that try to minimize an intraclass inertia with respect to constraints on the partition. The *k-means* algorithm, which assigns an object to a unique cluster, has been shown to be useful in various domains such as Internet of Things [1], time series [2], recommender systems [3, 4], among others. However, the obtained partition, called hard or crisp partition, is unable to express uncertainties regarding the class membership of an object. Such information is particularly interesting in case, for instance, of overlapped classes. Thus, modifications of the *k-means* algorithm have been proposed in order to generate a soft partition. The most popular extension corresponds to the *Fuzzy c-means* (FCM) that produces a probabilistic partition. It has been applied in many applications [5, 6, 7, 8]. Other variants, such as *Possibilistic c-means* and *Rough k-means*, use possibilities and rough set theories respectively to handle more precisely uncertainties. A more general variant, called *evidential c-means* (ECM) [9], is based on the theory of belief functions. It generates a credal partition that encompasses hard, probabilistic, and rough partitions [10]. The algorithm allows to obtain a rich representation of the uncertainties related to the data. As a consequence, it has been used in various applications [11, 12, 13] and several extensions of ECM have been proposed ever since. The RECM algorithm has been developed to handle dissimilarity data [14]. The ECMdd is a medoid-based variant of ECM [15]. The CCM method considers meta-clusters to reduce misclassification [16] and the DEC algorithm extends CCM by providing a dynamic edited framework [17]. Evidential clustering also comprises EVCLUS [18], a method that searches for a credal partition minimizing the discrepancy between the object pairwise distances and the conflict obtained by their mass functions. A faster optimization of EVCLUS has been proposed in [19].

Clustering algorithms create groups with no other information than the characteristics of the data. However, it has been shown that introducing some background knowledge in a clustering process can highly improve the clustering solution [20, 21, 22, 23]. Such methods, called semi-supervised clustering or constrained clustering, express prior information as constraints. The most famous types of constraints correspond to instance-level constraints: the must-link constraint, which specifies that two objects should be in the same class, the cannot-link constraint, which indicates that two objects are in different classes, and finally, the label constraint, which directly assigns an object into a class. Recently, various semi-supervised evidential clustering algorithms have been proposed for pairwise (i.e must-link and cannot-link) constraints [24, 25, 26, 27, 28] and for label constraints [29, 30, 27, 28]. The evidential framework is used to express imprecision for label constraints and allows for any type of instance-level constraints, as for evidential clustering, to generate a credal partition. However, the integration of instance-level constraints implies a greater optimization complexity. Thus, [26] proposes a new optimization scheme on CEVCLUS [25], a version of EVCLUS handling pairwise constraints. This method, named k-CEVCLUS, iteratively optimizes for each instance their mass function using a quadratic programming solver. k-CEVCLUS has been generalized by NN-EVCLUS [28] which uses a neural network trained to find the mass functions which minimize the difference between conflicts and the pairwise distances. The NN-EVCLUS algorithm copes with both pairwise and label constraints.

Although the pairwise constraints are more general, the label constraint is often available and provides more information. Thus, many *k-means* variants using label constraints have been proposed [20, 31, 29]. Amongst them, the SECM algorithm [29, 30] corresponds to the extension of ECM for fuzzy labels. Its goal is twofold: guide the clustering algorithm towards a better solution and take advantage of the richness of information available with credal partition to make decisions.

Similarly to any semi-supervised variants of *k-means* [20], the SECM algorithm adds a penalty term into the objective function in order to take into account the label constraints. Such penalty term complicates the criterion and leads to an increase in the computing time spent for its resolution. Thus, we propose to relax some constraints of the SECM objective function in order to create a new heuristic. The new algorithm, called SECM-h, increases the convergence speed of minimization while keeping a clustering solution close to the solution produced by the exact approach, i.e. SECM.

The rest of the paper is organized as follows. Section 2 recalls the necessary background on the theory of belief functions and the extensions of *k-means* leading to the evidential c-means. Then, the SECM algorithm is detailed. Section 3 depicts the SECM-h algorithm, a new approach to minimize SECM. The interest of the method is presented in Section 4 with some experiences on real data sets. Section 5 concludes the paper and gives some perspectives.

2. Preliminaries

Evidential clustering is based on the Dempster-Shafer theory or theory of belief functions. Then, to make this paper self-contained, a reminder of some important definitions and results of the Dempster-Shafer theory is presented in this section.

2.1. Theory of Belief Functions

In order to provide the belief function framework used in the following sections, we present the following definitions based on [32, 33].

Definition 1. *A* ***frame of discernment****,* $\Omega = \{\omega_1, \ldots, \omega_c\}$*, is said to be a set of all possible states that correspond to the interpretation of a finite propositional language.*

Let a proposition be represented by a variable ω which is assumed to take values in Ω, then the partial knowledge regarding such proposition can be represented by a *basic belief assignment* defined as follows:

Definition 2. *Let* $m : 2^\Omega \rightarrow [0, 1]$ *be a mass function. If* m *verifies:*

$$\sum_{A_j \subseteq \Omega} m(A_j) = 1. \tag{1}$$

then, m *is called* ***basic belief assignment*** *(*bba*).*

In order to interpret the mass functions, a major focus on the positive degree, i.e. $m(A_j) > 0$, of belief is performed which lead us to the following definition.

Definition 3. *Let $A_j \in \Omega$ such that $m(A_j) > 0$, then A_j is said to be a* ***focal set*** *of m.*

Special cases arise depending on which focal sets are actives [19]. As such, if focal sets only consist of singletons, then the mass function is said to be *Bayesian*. If the whole belief is allocated to a unique singleton, then the *bba* is considered as *certain*. Finally, if $m(\Omega) = 1$, then it represents a total uncertainty about the real state of ω.

Originally in [32], Equation (1) is constrained with $m(\emptyset) = 0$. Such mass function, called a *normal bba*, assumes that one state in Ω corresponds to the actual value of ω, leading to a closed-world assumption [33]. Inversely, if there exists the possibility that $\omega \notin \Omega$, then the empty set can be a *focal set*. This open-world assumption can be interpreted as the belief given to other states not included in Ω.

A *bba* can be transformed into a *normal bba* by transferring the empty set belief to all the other sets $A_j \subseteq \Omega$ [33]:

$$m^*(A_j) \triangleq \begin{cases} \frac{m(A_j)}{1-m(\emptyset)} & \text{if } A_j \neq \emptyset, \\ 0 & \text{otherwise.} \end{cases}$$

In [34], it has been shown that various measures can be associated with a given *bba*. Amongst them, the plausibility function $pl(A_j) : 2^\Omega \to [0,1]$ allows to quantify the amount of possibility that A_j includes the real state ω.

$$pl(A_j) \triangleq \sum_{A_\ell \subseteq \Omega, A_j \cap A_\ell \neq \emptyset} m(A_\ell) \quad \forall A_j \subseteq \Omega.$$

The probability function $BetP(\omega) : \Omega \to [0,1]$ provides from a *bba* a pignistic probability distribution:

$$BetP(\omega) \triangleq \sum_{\omega \in A_\ell} \frac{m^*(A_\ell)}{|A_\ell|} \quad \forall \omega \subseteq \Omega.$$

where $|A_\ell|$ denotes the cardinality of $A_\ell \subseteq \Omega$.

2.2. Evidential c-Means algorithm (ECM)

Evidential C-Means (ECM) is a clustering algorithm for object data introduced in [9] which is based on the concept of credal partition defined as follows:

Definition 4. *Let $\Omega = \{\omega_1, \cdots, \omega_c\}$ be the set of all possible clusters for the set of objects $\mathbf{X} = \{\mathbf{x_i} \in \mathbb{R}^p\}$ $\forall i = 1, .., n$ and m_i a mass function defined in Ω. Then, the collection $\mathbf{M} = (m_1, \cdots, m_n)^T \in \mathbb{R}^{n \times 2^c}$ is called the **credal partition** of $\mathbf{X}$.*

As discussed in [9], a credal partition is a generalization of well-known data partitions such as: hard, fuzzy, and possibilistic. It allows to represent the uncertainty and imprecision regarding the cluster membership of each object $\mathbf{x_i} \in \mathbf{X}$. ECM assumes that each cluster ω_k is represented by a prototype or barycenter $\mathbf{v}_k \in \mathbb{R}^p$ and the set of all prototypes is defined as the collection $\mathbf{V} = \{\mathbf{v}_k\}$. Since ECM is a credibilistic variant of Fuzzy C-Means (FCM) [35], then the evidential clustering problem has been stated in [9], [24] as finding the credal partition $\mathbf{M}$ that minimizes the following cost function:

$$J_{\mathrm{ECM}}(\mathbf{M}, \mathbf{V}, \mathbf{S}) = \sum_{i=1}^{n} \sum_{A_j \subseteq \Omega, A_j \neq \emptyset} |A_j|^{\alpha} m_{ij}^{\beta} d_{ij}^2 + \sum_{i=1}^{n} \delta^2 m_{i\emptyset}^{\beta}, \tag{2}$$

subject to

$$\sum_{A_j \subseteq \Omega, A_j \neq \emptyset} m_{ij} + m_{i\emptyset} = 1 \quad \forall i \in \{1, n\} \tag{3}$$

$$m_{ij} \geq 0 \quad \forall i \in \{1, n\}, \forall A_j \subseteq \Omega. \tag{4}$$

$$\det(\mathbf{S}_k) = 1 \qquad \forall k \in \{1, c\} \tag{5}$$

where $m_{ij} = m_i(A_j)$ corresponds to the mass of the object i allocated to the subset A_j, $m_{i\emptyset}$ defines more particularly the mass for the object i allocated to the empty set, $\alpha \geq 1$ is a parameter that allows controlling the degree of imprecision of the partition, $\beta > 1$ is used to penalize a high level of uncertainty on the partition, δ is a positive constant parameter that controls the importance given to the empty set, $\mathbf{S}_k$ is the covariance matrix of the k^{th} cluster, and d_{ij} is the Mahalanobis distance defined as:

$$d_{ij}^2 = \|\mathbf{x}_i - \mathbf{v}_j\|_{\overline{\mathbf{S}}_j}^2 = (\mathbf{x}_i - \mathbf{v}_j)^T \, \overline{\mathbf{S}}_j \, (\mathbf{x}_i - \mathbf{v}_j). \tag{6}$$

Let $\overline{\mathbf{v}}_j$ be the prototype for the subset A_j such that

$$\overline{\mathbf{v}}_j \triangleq \frac{1}{|A_j|} \sum_{l=1}^{c} s_{lj} \mathbf{v}_l \qquad \forall A_j \in \Omega,$$

with

$$s_{lj} = \begin{cases} 1 & \text{if } \omega_l \in A_j, \\ 0 & \text{otherwise.} \end{cases}$$

The set $\mathbf{S} = \{\mathbf{S}_1, \dots \mathbf{S}_c\}$ designates the covariance matrices for the clusters and $\overline{\mathbf{S}}_j$ represents the evidential covariance matrix for the j^{th} prototype. It is defined as:

$$\overline{\mathbf{S}}_j \triangleq \frac{1}{|A_j|} \sum_{\omega_k \in A_j} \mathbf{S}_k \qquad \forall A_j \in \Omega.$$

The optimal solution is obtained using a Gauss-Seidel type optimization method. First, $\mathbf{V}$ and $\mathbf{S}$ are assumed to be fixed, in order to obtain the solution of the constrained problem (2) with respect to $\mathbf{M}$. The optimality of $\mathbf{M}$ is obtained with

$$m_{ij} = \frac{|A_j|^{-\alpha/(\beta-1)} \, d_{ij}^{-2/(\beta-1)}}{\sum_{A_k \neq \emptyset} |A_k|^{-\alpha/(\beta-1)} \, d_{ik}^{-2/(\beta-1)} + \delta^{-2/(\beta-1)}} \quad \forall i = \{1, n\} \quad \forall A_j \neq \emptyset,$$

and

$$m_{i\emptyset} = 1 - \sum_{A_j \neq \emptyset} m_{ij} \quad \forall i = \{1, n\}.$$

Following the successive displacement optimization method, $\mathbf{M}$ and $\mathbf{S}$ are assumed to be fixed and the optimality of $\mathbf{V}$ is obtained as the solution of the linear system

$$\mathbf{H}\mathbf{V} = \mathbf{B}, \tag{7}$$

where $\mathbf{B} \in \mathbb{R}^{c \times p}$ is given by

$$\mathbf{B}_{\ell q} = \sum_{i=1}^{n} x_{iq} \sum_{A_j \neq \emptyset} |A_j|^{\alpha-1} \, m_{ij}^{\beta} s_{\ell j} \quad \forall \ell = \{1, c\} \quad \forall q = \{1, p\}, \tag{8}$$

and $\mathbf{H} \in \mathbb{R}^{c \times c}$ is given by:

$$\mathbf{H}_{\ell k} = \sum_{i=1}^{n} \sum_{A_j \neq \emptyset} |A_j|^{\alpha-2} \, m_{ij}^{\beta} s_{\ell j} s_{kj} \quad \forall k, \ell = \{1, c\}. \tag{9}$$

Finally, $\mathbf{M}$ and $\mathbf{V}$ are assumed to be fixed in order to obtain the optimal solution for (2) w.r.t. $\mathbf{S}$. To do so, the following change of variable is defined

$$\mathbf{\Sigma}_k \triangleq \sum_{i=1}^{n} \sum_{A_j \ni \omega_k} |A_j|^{\alpha-1} m_{ij}^{\beta} \left(\mathbf{x}_i - \overline{\mathbf{v}}_j\right) \left(\mathbf{x}_i - \overline{\mathbf{v}}_j\right)^T \quad \forall k = \{1, c\}. \tag{10}$$

Then, the optimal solution for $\mathbf{S}$ is given by

$$\mathbf{S}_k = \det(\mathbf{\Sigma}_k)^{1/p}\mathbf{\Sigma}_k^{-1}. \tag{11}$$

ECM provides sufficient conditions to find the optimal credal partition on object data. However, it does not take into account previous knowledge which is present in many real clustering applications. Such knowledge often takes the form of labeled data.

2.3. Semi-supervised ECM (SECM)

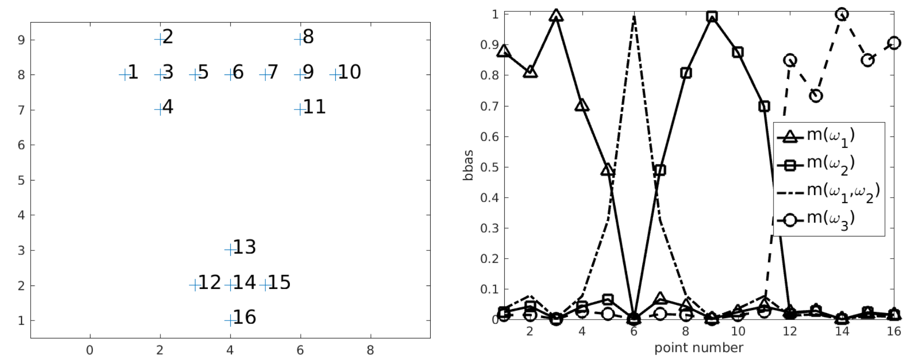
It is known that data labeling is usually carried out by experts. However, experts often face the problem of labeling objects for which they are not fully certain of the class they belong to. Then, in [30], a method to handle imprecise but certain labels is presented. An imprecise label on an object i is expressed with a subset $A_j \subset \Omega$, $|A_j| \geq 1$ such that the real class of $\mathbf{x}_i$ belongs to A_j. The consistency between this imprecise label subset A_j and the *bba* m_i assigned to object i is evaluated using the T_{ij} measure defined as follows:

$$T_{ij} \triangleq T_i\left(A_j\right) = \sum_{A_\ell \subseteq \Omega, A_j \cap A_\ell \neq \emptyset} \frac{|A_j \cap A_\ell|^{\frac{r}{2}}}{|A_\ell|^r} m_{i\ell}, \quad \forall i \in \{1, n\}, A_j \subseteq \Omega. \tag{12}$$

Such measure favors degrees of belief containing the constrained subset A_j or a part of it as well as subsets with low cardinalities. The $r \geq 0$ parameter controls the degree of precision expected for the object i: high values of r penalize subsets with high cardinalities whereas low values of r allow high value of T_{ij} for subsets with high cardinalities. If $r = 0$, T_{ij} corresponds to the plausibility that the object $\mathbf{x}_i$ belongs to A_j.

Notice that $T_{ij} = 1$ when belief is assigned for an object i that belongs to a cluster in the subset A_j. Inversely, $T_{ij} = 0$ when no belief is assigned for an object i to belong to a cluster in A_j, i.e., $x_i \notin A_j$.

To illustrate the behavior of the T_{ij} measure, let us consider DiamondK3, a synthetic data set composed of 15 points that needs to be divided into three groups (cf. Figure 1(a)). An execution of the ECM algorithm with $\alpha = 1$, $\beta = 2$, and $\delta^2 = 100$ provides us a credal partition, the main masses of which are presented in Figure 1(b). As it can be seen, the highest beliefs on the clusters ω_1 and ω_2 are correctly recovering the points 1 to 10 and the highest beliefs on the last cluster ω_3 are assigned to the points 12 to 15. Point 6, located between two classes, has all its mass allocated to the subset $\{\omega_1, \omega_2\}$.



(a) (b)

Figure 1: DiamondK3 data set (a) and a sample of the mass functions obtained using ECM (b).

Let us suppose that we know that object 6 belongs to ω_1 and object 13 belongs either to ω_1 or to ω_3, but not to ω_2. Such partial knowledge can be traduced into the following constraints: $\mathbf{x}_6 \in \{\omega_1\}$ and $\mathbf{x}_{13} \in \{\omega_1, \omega_3\}$. The T_{ij} measure allow us to quantify the degree of agreement between the constraint and the mass function obtained by ECM. The highest is the measure, the better the constraint is respected.

To better understand the concept, Table 1 presents a set of possible mass functions for the object 6 and the subsequent computation of $T_6(\{\omega_1\})$ using $r = 1$, and then with $r = 0$, i.e. $Pl_6(\{\omega_1\})$. The first case fully respects the constraint $\mathbf{x}_6 \in \{\omega_1\}$, giving $T_6(\{\omega_1\}) = Pl_6(\{\omega_1\}) = 1$ whereas case 4 totally ignores the constraints, giving $T_6(\{\omega_1\}) = Pl_6(\{\omega_1\}) = 0$. The second and the third case partially respect the constraint, since the bba is allocated to subsets containing ω_1. When $r = 1$, the more the cardinality of the subset, the lower the value of $T_6(\{\omega_1\})$. With the plausibility measure, any partial information gives $Pl_6(\{\omega_1\}) = 1$. This means that a partial respect of the constraint is considered as enough.

Table 1: Possible mass functions for object $\mathbf{x}_6$ constrained on $\{\omega_1\}$ and calculation of $T_i(A_j)$ with $r = 1$ and $Pl_i(A_j)$. Columns $\omega_{j,l}$ show the mass assigned to subset $\{\omega_j, \omega_l\}$.

$m_6(A_j)$	$\emptyset$	ω_1	ω_2	ω_3	$\omega_{1,2}$	$\omega_{1,3}$	$\omega_{2,3}$	Ω	$T_6(\{\omega_1\})$	$Pl_6(\{\omega_1\})$
case 1	0	1	0	0	0	0	0	0	1	1
case 2	0	0	0	0	1	0	0	0	1/2	1
case 3	0	0	0	0	0	0	0	1	1/3	1
case 4	0	0	1	0	0	0	0	0	0	0

Table 2 shows different possible mass function for the object 13. As it can be observed, the plausibility measure ensures the partial respect of the constraints. When $r = 1$, T_{ij} is null when no focal sets contains ω_1 and/or ω_3. Conversely, $Tij > 0$ when there exists degree of belief on a subset including at least ω_1 or ω_3. The measure favors subset with a low cardinality. For the same amount of subsets, for example in Table 2 $\{\omega_1, \omega_2\}$ and $\{\omega_1, \omega_3\}$, a higher value is given to subsets containing the most of classes in the constraint, i.e. here $\{\omega_1, \omega_3\}$.

Table 2: Possible mass functions for object $\mathbf{x}_{13}$ constrained on $\{\omega_1, \omega_3\}$ and calculation of T_{ij} with $r = 1$ and $Pl_i(A_j)$. Columns $\omega_{j,l}$ show the mass assigned to subset $\{\omega_j, \omega_l\}$.

$m_{13}(A_j)$	$\emptyset$	ω_1	ω_2	ω_3	$\omega_{1,2}$	$\omega_{1,3}$	$\omega_{2,3}$	Ω	$T_{13}(\{\omega_1, \omega_3\})$	$Pl_{13}(\{\omega_1, \omega_3\})$
case 1	0	1	0	0	0	0	0	0	1	1
case 2	0	0	0	0	0	1	0	0	$\sqrt{2}/2$	1
case 3	0	0	0	0	1	0	0	0	1/2	1
case 4	0	0	0	0	0	0	0	1	1/3	1
case 5	0	0	1	0	0	0	0	0	0	0

The T_{ij} measure is used to define the following penalty term:

$$J_s \triangleq \sum_{i=1}^{n} \sum_{A_j \subseteq \Omega, A_j \neq \emptyset} b_{ij}(1 - T_{ij}),$$

such that

$$b_{ij} = \begin{cases} 1 & \text{if } \mathbf{x}_i \text{ is constrained by } A_j, \\ 0 & \text{otherwise.} \end{cases}$$

The problem of semi-supervised evidential clustering is then defined as finding the 3-tuple $(\mathbf{M}, \mathbf{V}, \mathbf{S})$ that minimizes the following cost function

$$J_{SECM}(\mathbf{M}, \mathbf{V}, \mathbf{S}) = (1-\gamma)\frac{1}{2^c n} J_{ECM}(\mathbf{M}, \mathbf{V}, \mathbf{S}) + \gamma \frac{1}{s} J_s(\mathbf{M}), \qquad (13)$$

subject constraints (3)-(5). The s parameter is the number of constraints, $\frac{1}{2^c n}$ and $\frac{1}{s}$ are coefficients to balance the two terms [1] and $\gamma \in [0,1]$ the coefficient to control the tradeoff between the objective function of ECM and the constraints. In order to lighten the notations, we define ξ and χ such as:

$$\xi \triangleq (1-\gamma)\frac{1}{2^c n}, \qquad (14)$$

$$\chi \triangleq \gamma \frac{1}{s}, \qquad (15)$$

and

$$c_{\ell j} \triangleq \frac{|A_j \cap A_\ell|^{\frac{r}{2}}}{|A_j|^r}. \qquad (16)$$

Using (14)-(16), the objective function (13) can be rewritten as:

$$\begin{aligned} J_{SECM}(\mathbf{M}, \mathbf{V}, \mathbf{S}) = {} & \xi \left(\sum_{i=1}^{n} \sum_{A_j \neq \emptyset} |A_j|^\alpha m_{ij}^\beta d_{ij}^2 + \sum_{i=1}^{n} \delta^2 m_{i\emptyset}^\beta \right) \\ & + \chi \left(\sum_{i=1}^{n} \sum_{A_j \subseteq \Omega, A_j \neq \emptyset} b_{ij} \left(1 - \sum_{A_j \cap A_\ell \neq \emptyset} c_{\ell j} m_{i\ell} \right) \right) \end{aligned} \qquad (17)$$

When the β parameter is set to 2, the minimization problem in (17) becomes a quadratic optimization problem with linear constraints which is solved using classical methods [36, 37].

3. Heuristic Semi-supervised Evidential Clustering

3.1. Optimization scheme

The SECM problem discussed in the previous section can also be solved using a heuristic approach. Indeed, heuristic methods have been used in optimization

[1] Providing that the distances between objects and centroids are less or equal to 1. This can be done using a data normalization method, e.g. min-max scaling.

problems to improve the algorithm's execution time [20, 21]. Such methods usually result in a trade off between execution time and optimality, completeness and/or accuracy.

As discussed in section 2.3, the SECM problem has been solved using a Gauss-Seidel optimization method to find the 3-tuple $(\mathbf{V}, \mathbf{S}, \mathbf{M})$ that minimizes the cost function (13). Then, let us recall that the optimal centers of gravity $\mathbf{V}$ are found using (7) while the covariance matrix $\mathbf{S}$ are obtained using (11). The credal partition $\mathbf{M}$ can be found using both exact and heuristic methods which are discussed in the following sections.

3.2. Exact optimization

In order to obtain the credal partition $\mathbf{M}$ using an exact method, let $\mathbf{V}$ and $\mathbf{S}$ be constant and the β parameter of the cost function be set to 2. Let $\mathbf{m}_i \in \mathbb{R}^{1\times 2^c}$ be the mass for object $\mathbf{x}_i$ and the focal sets in the first term of (17) be rewritten in matrix form, $\mathbf{G}_i \in \mathbb{R}^{n2^c \times n2^c}$, using

$$G_{ij} \triangleq \begin{cases} \delta^2 & \text{if } A_j = \emptyset \\ |A_j|^\alpha d_{ij}^2 & \text{otherwise.} \end{cases}$$

Developing J_s, the second term in (17), we obtain

$$J_s = \sum_{i=1}^{n} \sum_{A_j \subseteq \Omega, A_j \neq \emptyset} b_{ij} - \sum_{i=1}^{n} \sum_{A_\ell \subseteq \Omega} m_{ij} (\sum_{A_j \subseteq \Omega} b_{ij\ell} c_{\ell j}),$$

with

$$b_{ij\ell} \triangleq \begin{cases} 1 & \text{if } \mathbf{x}_i \text{ is constrained to } A_j \text{ and } A_\ell \cap A_j \neq \emptyset, \\ 0 & \text{otherwise.} \end{cases}$$

Finally, the label constraints can also be written in matrix form $\mathbf{F}_i \in \mathbb{R}^{1\times 2^c}$ as:

$$F_{i\ell} \triangleq \begin{cases} 0 & \text{if } A_\ell = \emptyset \\ -\sum_{A_j \subseteq \Omega} b_{ij\ell} c_{\ell j} & \text{otherwise,} \end{cases}$$

and

$$f_i \triangleq \sum_{A_j} b_{ij}.$$

Then, the objective function (17) can be rewritten as follows:

$$J_{SECM}(\mathbf{M}) = \xi \sum_{i=1}^{n} \mathbf{m}_i^T \mathbf{G}_i \mathbf{m}_i + \chi \mathbf{F}_i^T \mathbf{m}_i + \chi f_i. \tag{18}$$

which can be solved using the well known sequential quadratic programming method [36, 37].

3.3. Heuristic Optimization

In order to improve algorithm execution time we propose a heuristic algorithm that relaxes constraint (4). It eases the optimization problem for the update of the mass m_{ij}. However, constraints (3) and (4) together imply that $0 \leq m_{ij} \leq 1 \ \forall i \in \{1, n\}, \forall A_j \subseteq \Omega$. Thus, removing constraint (4) allows $m_{ij} \in \mathbb{R}$. As it will be discussed later in this section, the case $m_{ij} < 0$ only appears on constrained objects $\mathbf{x}_i$, for subsets A_j violating the constraints. As a consequence, to respect constraint (3), the case $m_{ij'} > 1$ appears for subsets $A_{j'} \neq A_j$ respecting the label constraint. Since constraint (3) is respected and constraint (4) should be restored to comply with the definition (2) of a mass function, negative elements m_{ij} can be set to 0 and the negative quantity of belief can be used as a coefficient to equivalently lower the other masses. This is achieved using the following mass re-assignment:

$$m_{ij}^* = \begin{cases} \frac{m_{ij}}{1+a_i} & \text{if } m_{ij} > 0, \\ 0 & \text{otherwise,} \end{cases} \qquad \forall i = 1 \dots n, \forall A_j \subseteq \Omega \tag{19}$$

such that

$$a_i = - \sum_{A_j \subseteq \Omega} \min(m_{ij}, 0).$$

The optimal credibilistic partition can be found using the Lagrangian multiplier method.

$$\mathcal{L}(\mathbf{M}, \lambda_1, \dots, \lambda_n) = J_{SECM} - \sum_{i=1}^{n} \lambda_i \left(\sum_{A_j \subseteq \Omega, A_j \neq \emptyset} m_{ij} + m_{i\emptyset} - 1 \right).$$

The derivatives w.r.t. $m_{i,j}$ and $m_{i\emptyset}$ are:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial m_{ij}} &= \xi 2|A_j|^\alpha m_{ij} d_{ij}^2 - \chi \left(\sum_{A_j \cap A_\ell \neq \emptyset} c_{lj} b_{ij\ell} \right) - \lambda_i, \\
\frac{\partial \mathcal{L}}{\partial m_{i\emptyset}} &= 2\xi \delta^2 m_{i\emptyset} - \lambda_i.
\end{aligned}$$

Annulating the derivatives gives

$$m_{ij} = \frac{\lambda_i}{2\xi} \frac{1}{|A_j|^\alpha d_{ij}^2} + \frac{\chi \left(\sum_{A_j \cap A_\ell \neq \emptyset} c_{lj} b_{ij\ell} \right)}{2\xi |A_j|^\alpha d_{ij}^2}, \tag{20}$$

$$m_{i\emptyset} = \frac{\lambda_i}{2\xi \delta^2}. \tag{21}$$

Solving $\frac{\partial \mathcal{L}}{\partial \lambda_i} = 0$, we obtain:

$$\left(\sum_{A_j \subseteq \Omega, A_j \neq \emptyset} \frac{\lambda_i}{2\xi} \frac{1}{|A_j|^\alpha d_{ij}^2} + \frac{\chi \left(\sum_{A_j \cap A_\ell \neq \emptyset} c_{lj} b_{ij\ell} \right)}{2\xi |A_j|^\alpha d_{ij}^2} \right) + \frac{\lambda_i}{2\xi \delta^2} = 1,$$

$$\frac{\lambda_i}{2\xi} = \frac{1 - \sum_{A_j \subseteq \Omega, A_j \neq \emptyset} \chi \left(\sum_{A_j \cap A_l \neq \emptyset} c_{lj} b_{ij\ell} \right) \left(2\xi |A_j|^\alpha d_{ij}^2 \right)^{-1}}{\sum_{A_j \subseteq \Omega, A_j \neq \emptyset} \left(|A_j|^\alpha d_{ij}^2 \right)^{-1} + \delta^{-2}}.$$

Replacing $\frac{\lambda_i}{2\xi}$ in (20) and (21) enables to obtain the update equations:

$$m_{ij} = \frac{1 - \sum_{A_l \subseteq \Omega, A_l \neq \emptyset} \chi \left(\sum_{A_l \cap A_{l'} \neq \emptyset} c_{ll'} b_{ij\ell'} \right) \left(2\xi |A_l|^\alpha d_{il}^2 \right)^{-1}}{|A_j|^\alpha d_{ij}^2 \sum_{A_l \subseteq \Omega, A_l \neq \emptyset} \left(|A_l|^\alpha d_{il}^2 \right)^{-1} + \delta^{-2}} + \frac{\chi \left(\sum_{A_j \cap A_l \neq \emptyset} c_{lj} b_{ij\ell} \right)}{2\xi |A_j|^\alpha d_{ij}^2}, \tag{22}$$

and

$$m_{i\emptyset} = 1 - \sum_{A_j \subseteq \Omega, A_j \neq \emptyset} m_{ij}. \tag{23}$$

As discussed above, since constraint (4) has been relaxed and as it can be seen Equation (22), the update of the mass m_{ij} can be negative. This can occur only for constrained objects. Indeed, negative values only appear in the first term, when the second term of the numerator is higher than 0, which is only possible for some $b_{ij\ell'} = 1$, i.e. when there exists a constraint on $\mathbf{x}_i$. This behavior can be controlled by reducing the value of the coefficient χ.

The solution for the semi-supervised Evidential clustering problem is summarized in Algorithm 1.

Algorithm 1 $SECM_H$

Require: data $\mathbf{X}$, number of classes c and subsets $A_j \subseteq \Omega$

Ensure: credal partition $\mathbf{M}$, centroids $\mathbf{V}$, covariance matrices $\mathbf{S}$

  $k \leftarrow 0$

  Initialize $\mathbf{V}_k$ randomly, initialize $\mathbf{S}$ to identity matrices.

  repeat

    $k \leftarrow k + 1$

    Compute distances using (6).

    Update credibilistic partition $\mathbf{M}_k$ using (22) and (23).

    Correct $\mathbf{M}_k$ to respect contraints (4) using eq. (19).

    Update cluster centroids $\mathbf{V}_k$ using eq. (7).

    Update covariance matrix $\mathbf{S}_k$ using eq. (11).

  until $\|\mathbf{V}_k - \mathbf{V}_{k-1}\| < \varepsilon$

3.4. Complexity analysis

The complexity of $SECM_H$ is defined by the sum of the complexity of the different steps in the loop. The difference with ECM resides in the update of the credal partition. Let f be the number of focal elements such that $f = 2^c$ if all subsets are used. A reduction of this value can be achieved by privileging specific subsets, for instance, $f = c + 2$ when only singletons, Ω, and the empty set are selected.

- The computation of the Mahalanobis distance Equation (6) is performed in $O(p^2nf)$. If the covariance matrices are fixed to identity matrices, i.e.

a Euclidean distance has been chosen, then the complexity is decreased to $O(pnf)$.

- The complexity for update of the credal partition Equations (22) and (23) is $O(nf)$. The correction using Equation (19) requires at the worst case nf operations. The overall complexity of the three equations is then $O(nf)$. This also corresponds to the complexity of the update of the credal partition for the ECM algorithm [15].
- To compute the update of the cluster centroids $\mathbf{V}_k$ Equation (7), it is first necessary to compute matrices $\mathbf{B}$, Equation (8), which requires $cpnf$ operations and $\mathbf{H}$, Equation (9), which requires c^2nf operations. Since our implementation uses a QR decomposition, its implies a complexity of $O(c^3)$ if $c \leq p$, $O(p^3)$ otherwise. The resulting complexity of the overall sub-steps is $O(cpnf + c^2nf + \min(c,p)^3)$.
- Finally, the complexity of the covariance matrix, Equation (11), that is used only when adaptive distances are chosen, is $O(cp^3)$ due to the computation of the inverse of $\mathbf{\Sigma}_k$. The computation of $\mathbf{\Sigma}_k$, Equation (10), requires cp^2nf operations.

As a result, the $SECM_H$ algorithm using adaptive distances has a complexity of $O(kc^2nf + k\min(c,p)^3 + kcp^3 + kcp^2nf)$ where k represents the number of iterations reached before convergence. When using a Euclidean distance, the complexity is $O(kcpnf + kc^2nf + k\min(c,p)^3)$. Considering that $n \gg p$ and $n \gg c$, the complexity for both distances becomes $O(knf + k\min(c,p)^3)$.

When the exact optimization is used to update of credal partition, i.e. a sequential quadratic programming method on Equation (17), the complexity is $O(n^3f^3)$[37]. Thus, the overall complexity of $SECM$ with an adaptive distance using an exact optimization (named $SECM_E$) is $O(kp^2nf + kn^3f^3 + kc^2nf + k\min(c,p)^3 + kcp^2nf)$. Considering that $n \gg p$ and $n \gg c$, the complexity becomes $O(kn^3f^3 + k\min(c,p)^3)$.

4. Results

4.1. Experimental protocol

To illustrate the effectiveness of the proposed clustering method, we carried out a series of experiments using data sets coming from the UCI repository [38]. The characteristics of the data sets are shown in Table 3.

Table 3: A quick view of UCI data sets involved.

Name	# objects	# attributes	# classes	# objects per classes
Column	310	6	3	{60,150,100}
Ionosphere	351	34	2	{225, 126}
Iris	150	4	3	{50, 50, 50}
LettersIJL	227	16	3	{81, 72, 74}
Wdbc	5569	30	2	{357,212}
Wine	178	13	3	{59, 71, 48}

Real labels are known for all data sets. LettersIJL refers to the Letters data set where only three classes are kept and the number of instances is reduced by randomly selecting 10% of them per class, as it has been done in [22]. Since Column, Wdbc and Wine are a data sets characterized by great variations of scale amongst the attributes, we use a standardization method to avoid unequal contributions of the features during the clustering process. The Mahalanobis distance has been used for all data sets except for Column, Wdbc and Wine that use the Euclidean distance as they have spherical shaped groups.

The performance of the evidential clustering method is measured using the Adjusted Rand Index (ARI) [39] that takes values in $[0, 1]$. The ARI is based on the pairwise comparisons of the crisp partition $\mathbf{P}$ with a second crisp partition $\mathbf{P}^*$. A perfect match between $\mathbf{P}$ and $\mathbf{P}^*$ is reflected by the ARI equal to 1. In our context, the $\mathbf{P}$ partition is obtained by first converting the credal partition $\mathbf{M}$ into a fuzzy partition $\mathbf{U}$ using the pignistic transformation. Then, the objects are assigned to clusters with the maximum of pignistic probability.

In order to determine the closeness of credal partitions, Denœux [10] proposed a Credal Rand Index (CRI), where CRI $\in [0, 1]$ and CRI=1 means that the compared partitions are identical. More precisely, we employ the CRI related to the similarity index with a Jousselme's distance.

In the framework of semi-supervised clustering, we also assess the performance of the clustering methods using well-known classification measures such as Precision, Recall, and F_1 score. These measures are computed as in [40] and generalized for multiclass. The precision quantifies from the number of objects predicted in a class how many are really in that class, whereas the recall indicates the fraction of objects in a class that is correctly assigned in that class. The F_1 score is the harmonic mean of the precision and the recall.

The $SECM_E$ and $SECM_H$ algorithms take into account prior information in the form of label constraints which may be imprecise. First, a percentage of the real labels of the data sets are selected to obtain the set $\mathcal{L}_s$ of hard labeled instances with no imprecision, also called *single set*. This percentage varies from 0 to 30%. Note that given a percentage p_1 allowing to obtain $\mathcal{L}_{s_{p_1}}$, the set $\mathcal{L}_{s_{p_2}}$ of a higher percentage $p_2 > p_1$ is created such that $\mathcal{L}_{s_{p_1}} \subset \mathcal{L}_{s_{p_2}}$ in order to better illustrate the interest of adding constraints. A *single set* is only composed of hard and clean labels. However, in a real application, these labels can be noisy. Thus, we introduced 20% of noise into the $\mathcal{L}_s$ set to form a hard *noisy label set* referred to as $\mathcal{L}_n$. Finally, the imprecision on the class membership of an object is simulated by the concatenation of $\mathcal{L}_s$ and $\mathcal{L}_n$, leading to the *double set* $\mathcal{L}_d$ that contains 20% of objects constrained to two classes. Notice that $\mathcal{L}_d$ is a set of imprecise labels but not noisy since for every labeled object, the real class is present in $\mathcal{L}_d$. The generation of the different sets of labels is resumed in Figure 2.

The evidential clustering algorithms $SECM_E$ and $SECM_H$ are compared with the most similar semi-supervised clustering algorithms that provide a hard partition (SKMEANS [31]) and a fuzzy partition (SFCM [20, 23]). Default parameters for the evidential methods are $\alpha = 1$, $\beta = 2$, $\delta^2 = 100$ and $\gamma = 0.5$. The exponent α is set to 1 as [9] to allow enough imprecision in the credal partition. The coefficient of fuzzification β is equivalently fixed for SFCM on the fuzzy partition. As discussed in [9], setting the δ parameter is difficult since

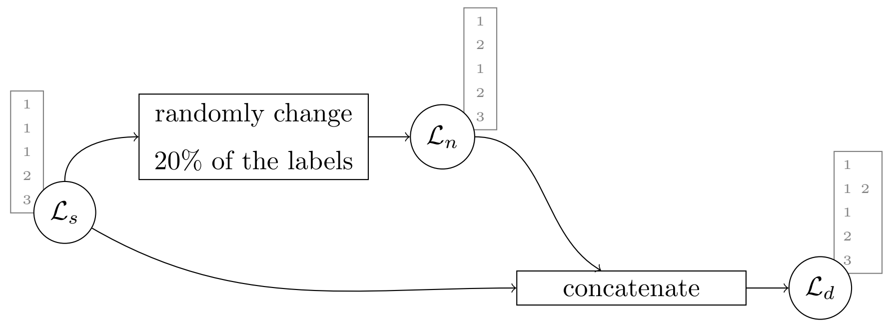


Figure 2: Generation of noisy and double labels. Gray numbers represent an example of the sets given three classes $\{1, 2, 3\}$.

it strongly depends on the data. In this work, we set δ parameter to a high value as no outliers exist in the data sets. The parameter γ, which controls the tradeoff between constraints and the data structure has been tested with values varying from 0.1 to 0.7. It is set by default for SFCM, $SECM_E$, and $SECM_H$ to 0.5 in order to be balanced.

The imprecise information contained in the *double set* $\mathcal{L}_d$ is exploited by SECM using constraints on subsets. However, this information cannot be taken by SKMEANS and must be transformed into uncertain information for SFCM. Indeed, SFCM only allows objects to be constrained to single classes with degrees of probability. Thus, we chose to set a probability of $\{0.5, 0.5\}$ for each object having imprecision between two classes. Note that the meaning of the information is slightly different as it represents precise information, although uncertain [41].

Since SKMEANS, SFCM, and SECM are semi-supervised variants of k-means, they all need a random initialization of the centroids and labels as prior information. With the view to making performance comparisons, the same initializations are applied for each algorithm and the same set of labels are used. An experiment consists of 100 trials of an algorithm. Results present the average and confidence interval measures (as the constraint sets are different for each trial). A trial is characterized by 5 executions of the algorithm with different initialization of the centroids. The final partition kept is the one with the minimum objective function. The experiments were performed on an Intel(R) Xeon(R) CPU E5-2670 v2 @ 2.50 GHz.

4.2. Influence of the γ parameter

The γ parameter is a coefficient set between 0 and 1 that allows giving more or less importance to the respect of the constraints. Tables 4 and 5 present several values of γ tested with $SECM_E$ and $SECM_H$ using 30% of *single* or *doble* constraints. The results correspond to the average of the ARI on the whole data sets and the average of the ARI on unconstrained objects only (ARIu).

Table 4: Influence of the γ parameter with 30% of constraints and $r = 1$. Standard deviations are shown in brackets, confidence intervals are between 0.00 and 0.01.

		$\mathcal{L}_s$ set				$\mathcal{L}_d$ set			
		$SECM_H$		$SECM_E$		$SECM_H$		$SECM_E$	
		ARI	ARIu	ARI	ARIu	ARI	ARIu	ARI	ARIu
Column	0.1	0.50 [0.04]	0.39 [0.05]	0.52 [0.03]	0.40 [0.03]	0.45 [0.05]	0.36 [0.05]	0.48 [0.04]	0.39 [0.04]
	0.3	0.57 [0.03]	0.43 [0.04]	0.58 [0.03]	0.44 [0.04]	0.54 [0.03]	0.42 [0.04]	0.54 [0.03]	0.43 [0.03]
	0.5	0.58 [0.03]	0.44 [0.04]	0.58 [0.03]	0.44 [0.04]	0.55 [0.03]	0.43 [0.04]	0.54 [0.03]	0.43 [0.03]
	0.7	0.58 [0.03]	0.44 [0.04]	0.57 [0.03]	0.43 [0.04]	0.55 [0.03]	0.43 [0.04]	0.54 [0.03]	0.43 [0.03]
Ionosphere	0.1	0.71 [0.04]	0.62 [0.05]	0.71 [0.03]	0.62 [0.05]	0.70 [0.03]	0.62 [0.04]	0.69 [0.03]	0.61 [0.04]
	0.3	0.67 [0.04]	0.55 [0.05]	0.66 [0.04]	0.54 [0.05]	0.65 [0.03]	0.56 [0.05]	0.64 [0.04]	0.55 [0.05]
	0.5	0.58 [0.04]	0.44 [0.05]	0.57 [0.04]	0.43 [0.05]	0.57 [0.04]	0.46 [0.05]	0.56 [0.04]	0.44 [0.05]
	0.7	0.54 [0.04]	0.38 [0.05]	0.52 [0.04]	0.36 [0.05]	0.53 [0.04]	0.41 [0.06]	0.51 [0.04]	0.38 [0.05]
Iris	0.1	0.92 [0.03]	0.89 [0.05]	0.93 [0.03]	0.90 [0.05]	0.92 [0.04]	0.89 [0.05]	0.92 [0.04]	0.89 [0.05]
	0.3	0.92 [0.03]	0.89 [0.05]	0.93 [0.03]	0.89 [0.04]	0.92 [0.03]	0.89 [0.05]	0.92 [0.04]	0.89 [0.06]
	0.5	0.92 [0.03]	0.89 [0.05]	0.93 [0.03]	0.90 [0.05]	0.92 [0.03]	0.89 [0.05]	0.91 [0.04]	0.88 [0.06]
	0.7	0.92 [0.03]	0.89 [0.05]	0.93 [0.03]	0.90 [0.05]	0.92 [0.04]	0.89 [0.05]	0.92 [0.03]	0.89 [0.05]

The constraint set (i.e. $\mathcal{L}_s$ or $\mathcal{L}_d$) and the algorithm (i.e. $SECM_E$ or $SECM_H$) does not seem to be parameters that influence the choice of γ. We can also remark that no significant differences in behavior exist between the ARI and the ARIu.

Table 5: Influence of the γ parameter with 30% of constraints and $r = 1$. Standard deviations are shown in brackets, confidence intervals are between 0.00 and 0.01.

		$\mathcal{L}_s$ set				$\mathcal{L}_d$ set			
		$SECM_H$		$SECM_E$		$SECM_H$		$SECM_E$	
		ARI	ARIu	ARI	ARIu	ARI	ARIu	ARI	ARIu
LettersIJL	0.1	0.65 [0.05]	0.56 [0.06]	0.65 [0.05]	0.57 [0.06]	0.63 [0.06]	0.55 [0.07]	0.63 [0.06]	0.55 [0.07]
	0.3	0.72 [0.05]	0.62 [0.06]	0.72 [0.05]	0.62 [0.06]	0.69 [0.05]	0.60 [0.07]	0.69 [0.05]	0.60 [0.07]
	0.5	0.75 [0.05]	0.65 [0.06]	0.75 [0.05]	0.65 [0.07]	0.72 [0.06]	0.63 [0.07]	0.72 [0.06]	0.63 [0.07]
	0.7	0.76 [0.05]	0.66 [0.07]	0.77 [0.05]	0.68 [0.07]	0.73 [0.06]	0.65 [0.08]	0.74 [0.06]	0.65 [0.08]
Wdbc	0.1	0.70 [0.01]	0.69 [0.02]	0.70 [0.01]	0.69 [0.02]	0.70 [0.01]	0.69 [0.02]	0.70 [0.01]	0.69 [0.02]
	0.3	0.72 [0.01]	0.69 [0.02]	0.73 [0.01]	0.69 [0.02]	0.72 [0.01]	0.69 [0.02]	0.72 [0.01]	0.69 [0.02]
	0.5	0.76 [0.02]	0.69 [0.02]	0.76 [0.02]	0.70 [0.03]	0.74 [0.02]	0.69 [0.02]	0.75 [0.02]	0.69 [0.03]
	0.7	0.79 [0.02]	0.70 [0.03]	0.79 [0.02]	0.72 [0.03]	0.77 [0.02]	0.70 [0.03]	0.77 [0.02]	0.71 [0.03]
Wine	0.1	0.91 [0.01]	0.89 [0.03]	0.91 [0.01]	0.90 [0.03]	0.91 [0.02]	0.89 [0.03]	0.91 [0.01]	0.90 [0.03]
	0.3	0.92 [0.02]	0.88 [0.03]	0.91 [0.02]	0.88 [0.03]	0.91 [0.02]	0.88 [0.03]	0.91 [0.02]	0.88 [0.03]
	0.5	0.92 [0.02]	0.88 [0.03]	0.92 [0.03]	0.88 [0.04]	0.91 [0.02]	0.88 [0.04]	0.91 [0.02]	0.88 [0.03]
	0.7	0.92 [0.03]	0.88 [0.04]	0.92 [0.03]	0.88 [0.04]	0.91 [0.03]	0.88 [0.04]	0.91 [0.03]	0.88 [0.04]

It can be observed that the best γ value mostly depends on the data set. Ionosphere, which has an extended region of overlapped classes, shows better performances with a low value of γ whereas LettersIJL, that has more prolate ellipsoids, has better results with a high value of γ.

Two schemes are possible when applying a SECM: 1) the overall structure of the data is already well defined with the unlabeled data and only overlapped areas between several clusters need the constraints to be less fuzzy. 2) the global structure found with unlabeled data is not the desired solution and the constraints must help to lead the algorithm towards this solution. In the first case, the γ value should be low to preserve the discovered structure. In the second case, the γ needs to be large enough to force the global structure towards a new different one. For the rest of the paper, we suppose that no prior information is available regarding the characteristics of the data sets. Thus, γ is fixed to 0.5.

4.3. Interest of the constraints

Adding constraints allows either to guide a clustering algorithm towards a desired solution or to refine a partition solution. To show the first behavior, let us consider a synthetic data set named GaussK2 that needs to be classified into two groups. The data points with ground truth class labels are shown Figure 3.

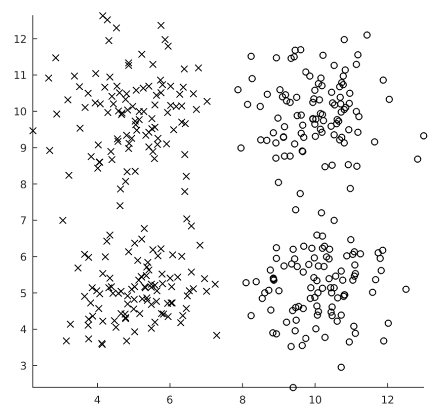
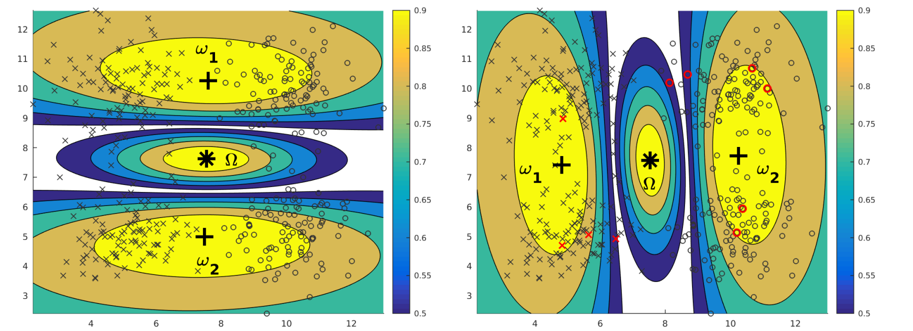


Figure 3: GaussK2 data set. Symbols represents the ground truth class labels.

The ECM algorithm, with $\alpha = 1$, $\beta = 2$, $\delta^2 = 100$, and a Mahalanobis distance separates data with a vertical or an horizontal frontier depending on the centroids initialization. A credal partition obtained with ECM is presented in Figure 4 (a). Since it does not correspond to the true partition, the ARI is equal to 0.0025. However, an execution of SECM with only 10 constraints (cf. Figure 4 (b)) allows to obtain a vertical boundary and an ARI equal to 1.

The interest of label constraints on real data sets is summarized in Table 6 using the precision (P), the recall (R), the F_1 score, and the Adjusted Rand Index (ARI).

As it can be observed, there exist little differences between the precision and the recall except for Columns that is characterized by unbalanced classes. The low recall on Column when no constraint exists means that the clustering algorithm had difficulties, for each class, to correctly identify objects inside. As



(a) (b)

Figure 4: Contour surfaces of the credal partition obtained by (a) ECM and (b) SECM. Centroids for singletons and the universe are designed by crosses and stars. Label constraints are highlighted in red.

shown Table 6, the gap between precision and recall is reduced with constraints and both of the measures are improved. We can conclude that label constraints help to better classify objects. In terms of performances, few differences are also observed between $SECM_H$ and $SECM_E$. Finally, the F_1 score and the ARI follow the same trend: the more constraints are added, the better the accuracy. Experiments were also conducted on the $\mathcal{L}_d$ set and the results lead to a similar analysis.

4.4. r *parameter*

The r parameter controls for labels constraints the penalization given to subsets with high cardinalities. Two values are tested: $r = 1$, which is the default setting of $SECM$, and $r = 0$, which is identical to replacing the T_{ij} terms by plausibilities. Thus, in the following experiments, this version is referred to as $SECMpl$. Experiments are conducted on the $\mathcal{L}_s$, $\mathcal{L}_d$ and $\mathcal{L}_n$ sets for a various percentage of constraints. Results are presented Table 7.

As expected, best performances are obtained with the most informative set, i.e. the $\mathcal{L}_s$ set, and the lower performances a retrieved by the *noisy set* $\mathcal{L}_n$. In the details, we can observe that for the $\mathcal{L}_s$ and the $\mathcal{L}_d$ sets, $SECM$ works better for Column and Iris whereas $SECMpl$ find better results for Ionosphere.

Table 6: Interest of the constraints using $\mathcal{L}_s$ set, $\gamma = 0.5$ and $r = 1$. Standard deviations are shown in brackets, confidence intervals are between 0.00 and 0.02.

		$SECM_H$				$SECM_E$			
		P	R	F_1	ARI	P	R	F_1	ARI
Column	0	0.65	0.48	0.52	0.23 [0.00]	0.65	0.48	0.52	0.23 [0.00]
	10	0.76	0.73	0.74	0.43 [0.04]	0.75	0.71	0.72	0.41 [0.05]
	20	0.81	0.79	0.79	0.51 [0.03]	0.81	0.79	0.79	0.51 [0.03]
	30	0.84	0.82	0.83	0.58 [0.03]	0.84	0.82	0.83	0.58 [0.03]
Ionosphere	0	0.69	0.56	0.55	0.01 [0.05]	0.69	0.56	0.55	0.01 [0.05]
	10	0.87	0.87	0.86	0.53 [0.06]	0.86	0.86	0.86	0.51 [0.06]
	20	0.87	0.87	0.87	0.54 [0.05]	0.87	0.87	0.86	0.53 [0.04]
	30	0.89	0.88	0.88	0.58 [0.04]	0.88	0.88	0.88	0.57 [0.04]
Iris	0	0.87	0.86	0.86	0.67 [0.01]	0.87	0.86	0.86	0.67 [0.01]
	10	0.94	0.93	0.93	0.82 [0.07]	0.94	0.93	0.93	0.83 [0.07]
	20	0.97	0.96	0.96	0.90 [0.05]	0.97	0.97	0.97	0.90 [0.05]
	30	0.97	0.97	0.97	0.92 [0.03]	0.98	0.97	0.97	0.93 [0.03]
LettersIJL	0	0.56	0.50	0.49	0.07 [0.04]	0.56	0.50	0.49	0.07 [0.04]
	10	0.83	0.82	0.82	0.56 [0.08]	0.83	0.83	0.82	0.56 [0.08]
	20	0.88	0.88	0.88	0.68 [0.06]	0.88	0.88	0.88	0.68 [0.06]
	30	0.91	0.91	0.91	0.75 [0.05]	0.91	0.91	0.91	0.75 [0.05]
Wdbc	0	0.92	0.92	0.92	0.69 [0.00]	0.92	0.92	0.92	0.69 [0.00]
	10	0.93	0.93	0.93	0.72 [0.01]	0.93	0.93	0.93	0.73 [0.02]
	20	0.93	0.93	0.93	0.74 [0.02]	0.93	0.93	0.93	0.75 [0.02]
	30	0.94	0.94	0.94	0.76 [0.02]	0.94	0.94	0.94	0.76 [0.02]
Wine	0	0.96	0.96	0.96	0.88 [0.01]	0.96	0.96	0.96	0.88 [0.01]
	10	0.96	0.96	0.96	0.87 [0.04]	0.96	0.96	0.96	0.88 [0.03]
	20	0.97	0.97	0.97	0.90 [0.03]	0.97	0.96	0.96	0.89 [0.03]
	30	0.97	0.97	0.97	0.92 [0.02]	0.97	0.97	0.97	0.92 [0.03]

To resume, the best performances mostly depends on the data sets.

On the contrary, $SECMpl$ has most of the time better performances than $SECM$ using the $\mathcal{L}_n$ set. Indeed, $r = 0$ let for constrained objects equal possibilities on subsets with various cardinalities. If the label corresponds to a noise and is in contradiction with the data structure, the algorithm will favor subsets with high cardinalities, i.e. subsets that include both the label and the cluster guessed with the data structure.

Table 7: Influence of the r parameter, $\gamma = 0.5$. Average ARI and its standard deviation in brackets are shown, confidence intervals are between 0 and 0.03.

		$\mathcal{L}_s$ set		$\mathcal{L}_d$ set		$\mathcal{L}_n$ set	
		$SECM_H$	$SECMpl_H$	$SECM_H$	$SECMpl_H$	$SECM_H$	$SECMpl_H$
Column	0	0.23 [0.00]	0.23 [0.00]	0.23 [0.00]	0.23 [0.00]	0.23 [0.00]	0.23 [0.00]
	10	**0.43** [0.04]	0.28 [0.02]	**0.42** [0.04]	0.27 [0.01]	**0.37** [0.04]	0.25 [0.02]
	20	**0.51** [0.03]	0.37 [0.04]	**0.49** [0.04]	0.34 [0.03]	**0.41** [0.04]	0.28 [0.03]
	30	**0.58** [0.03]	0.48 [0.04]	**0.55** [0.03]	0.43 [0.04]	**0.43** [0.03]	0.33 [0.03]
Ionosphere	0	0.01 [0.05]	0.01 [0.05]	0.01 [0.05]	0.01 [0.05]	0.01 [0.05]	0.01 [0.05]
	10	0.53 [0.06]	**0.57** [0.04]	0.54 [0.06]	**0.57** [0.04]	0.22 [0.13]	**0.41** [0.10]
	20	0.54 [0.05]	**0.59** [0.04]	0.55 [0.04]	**0.58** [0.04]	0.32 [0.07]	**0.40** [0.06]
	30	0.58 [0.04]	**0.61** [0.04]	0.57 [0.04]	**0.60** [0.04]	0.35 [0.04]	**0.38** [0.04]
Iris	0	0.67 [0.01]	0.67 [0.01]	0.67 [0.01]	0.67 [0.01]	0.67 [0.01]	0.67 [0.01]
	10	**0.82** [0.07]	0.77 [0.07]	**0.82** [0.07]	0.76 [0.07]	0.51 [0.14]	**0.62** [0.08]
	20	**0.90** [0.05]	0.86 [0.06]	**0.89** [0.05]	0.85 [0.07]	0.58 [0.10]	**0.61** [0.08]
	30	**0.92** [0.03]	0.89 [0.05]	**0.92** [0.03]	0.88 [0.06]	0.59 [0.08]	0.58 [0.07]
Letters	0	0.07 [0.04]	0.07 [0.04]	0.07 [0.04]	0.07 [0.04]	0.07 [0.04]	0.07[0.04]
	10	0.56 [0.08]	0.55 [0.08]	**0.54** [0.08]	0.52 [0.09]	0.38 [0.09]	0.39 [0.10]
	20	0.68 [0.06]	0.67 [0.05]	0.65 [0.06]	0.65 [0.06]	0.47 [0.08]	0.48 [0.08]
	30	0.75 [0.05]	0.74 [0.05]	0.72 [0.06]	0.71 [0.05]	0.51 [0.06]	0.51 [0.05]
Wdbc	0	0.69 [0.00]	0.69 [0.00]	0.69 [0.00]	0.69 [0.00]	0.69 [0.00]	0.69 [0.00]
	10	0.72 [0.01]	0.72 [0.01]	0.72 [0.01]	0.71 [0.01]	0.67 [0.02]	0.66 [0.01]
	20	0.74 [0.02]	0.75 [0.02]	0.73 [0.01]	0.74 [0.01]	0.70 [0.02]	0.68 [0.02]
	30	0.76 [0.02]	0.76 [0.02]	0.74 [0.02]	0.75 [0.02]	0.71 [0.02]	0.71 [0.02]
Wine	0	0.88 [0.02]	0.88 [0.02]	0.88 [0.02]	0.88 [0.02]	0.88 [0.02]	0.88 [0.02]
	10	0.87 [0.04]	**0.90** [0.02]	0.87 [0.03]	**0.90** [0.02]	0.79 [0.04]	**0.84** [0.02]
	20	0.90 [0.03]	0.91 [0.02]	0.89 [0.03]	**0.91** [0.02]	0.77 [0.03]	**0.79** [0.02]
	30	0.92 [0.02]	0.93 [0.02]	0.91 [0.02]	0.92 [0.02]	0.73 [0.03]	0.74 [0.02]

4.5. Credal partition analysis

Experiments conducted on previous subsection have shown that $SECM_H$ and $SECM_E$ have similar performances. The performance has been evaluated using the ARI measure, after a transformation of the credal partition into a hard partition. In order to study more precisely the difference between the $SECM_H$ and $SECM_E$ using similar parameters, the closeness between credal partitions is computed with the CRI. Table 8 presents the results on the $\mathcal{L}_s$ set. It can be concluded that there exists no significant difference between $SECM_H$ and $SECM_E$.

Table 8: CRI (mean[std]) between $SECM_H$ and $SECM_E$ using $\mathcal{L}_s$ set, $\gamma = 0.5$ and $r = 1$.

data set	10	20	30
Column	0.970[0.004]	0.961[0.004]	0.963[0.005]
Ionosphere	0.988[0.002]	0.987[0.002]	0.987[0.002]
Iris	0.989[0.005]	0.989[0.007]	0.988[0.009]
Letters	0.998[0.001]	0.995[0.001]	0.988[0.002]
Wdbc	0.996[0.000]	0.994[0.001]	0.991[0.001]
Wine	0.993[0.001]	0.997[0.001]	0.986[0.002]

4.6. Optimization analysis

A study concerning the optimization performances of $SECM_E$ and $SECM_H$ is performed. Table 9 presents the average and the confidence interval for the objective function, the penalization J_s, and the CPU time in seconds obtained by the two algorithms using a various number of constraints with the $\mathcal{L}_s$ set.

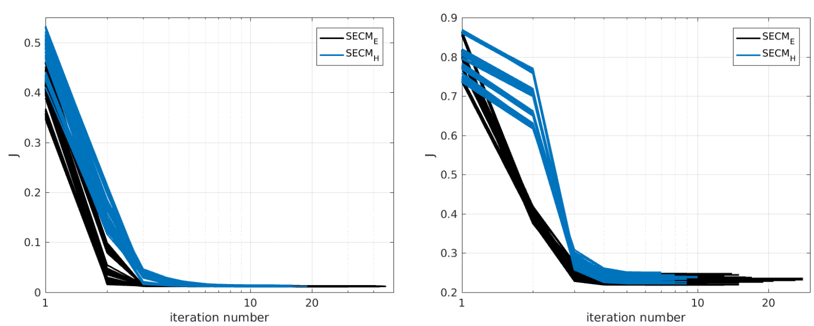
Results clearly show that the penalty term value and the objective function value are minimal with $SECM_E$. However, the difference between the objective function of $SECM_E$ and $SECM_H$ can be considered as insignificant since the two algorithms give equivalent clustering performances. On the other hand, the computation time of $SECM_H$ is at least two times lower than $SECM_E$.

The same experiments have been conducted on the $\mathcal{L}_d$ set. Although the CPU time is slightly larger than for the $\mathcal{L}_s$ set, the same interpretations can be deduced: $SECM_H$ is faster than $SECM_E$ and $SECM_E$ achieves the lowest objective function values.

Figure 5 illustrates the objective function values during the process of 100 execution of SECM on the $\mathcal{L}_s$ set for Iris and Wine. Similar results are obtained with the other data sets. As it can be remarked, the iteration number is lower for the $SECM_H$ algorithm and the minimum values of the objective function are reached by the $SECM_E$ algorithm.

Table 9: Analysis of the Optimization performances of SECM using $\mathcal{L}_s$ set, $\gamma = 0.5$ and $r = 1$.

		$SECM_H$			$SECM_E$		
		J_{SECM}	J_s $(\times 10^{-3})$	CPU (s)	J_{SECM}	J_s $(\times 10^{-3})$	CPU (s)
Column	0	0.0 ± 0.0	0.0 ± 0.0	0.30 ± 0.01	0.0 ± 0.0	0.0 ± 0.0	0.30 ± 0.01
	10	110.1 ± 1.4	61.4 ± 2.3	**0.10** ± 0.01	86.6 ± 0.9	**1.1** ± 0.6	5.31 ± 0.64
	20	124.7 ± 0.9	73.4 ± 1.5	**0.08** ± 0.00	104.1 ± 0.7	**6.8** ± 0.8	1.97 ± 0.14
	30	135.9 ± 0.9	81.9 ± 1.3	**0.11** ± 0.00	118.3 ± 0.8	**14.6** ± 0.8	1.42 ± 0.04
Ionosphere	0	0.2 ± 0.0	0.0 ± 0.0	2.56 ± 0.2	0.2 ± 0.0	0.0 ± 0.0	2.56 ± 0.20
	10	237.1 ± 0.5	1.6 ± 0.2	**1.03** ± 0.03	236.6 ± 0.5	**0.0** ± 0.0	2.57 ± 0.07
	20	252.7 ± 0.8	13.0 ± 0.5	**0.78** ± 0.01	251.6 ± 0.7	**7.0** ± 0.4	1.69 ± 0.04
	30	266.2 ± 0.8	29.6 ± 0.9	**0.69** ± 0.01	265.2 ± 0.8	**24.9** ± 0.8	1.45 ± 0.03
Iris	0	0.0 ± 0.0	0.0 ± 0.0	1.94 ± 0.06	0.0 ± 0.0	0.0 ± 0.0	1.94 ± 0.06
	10	10.8 ± 0.1	0.7 ± 0.2	**0.95** ± 0.05	10.4 ± 0.0	**0.0** ± 0.0	3.27 ± 0.14
	20	11.8 ± 0.1	0.9 ± 0.2	**0.63** ± 0.03	11.3 ± 0.1	**0.0** ± 0.0	2.26 ± 0.09
	30	12.5 ± 0.1	1.1 ± 0.2	**0.51** ± 0.02	12.0 ± 0.1	**0.0** ± 0.0	1.92 ± 0.07
LettersIJL	0	0.2 ± 0.0	0.0 ± 0.0	7.61 ± 0.63	0.2 ± 0.0	0.0 ± 0.0	7.61 ± 0.63
	10	420.5 ± 1.4	25.4 ± 1.3	**3.01** ± 0.22	415.4 ± 1.3	**0.3** ± 0.1	7.07 ± 0.78
	20	463.0 ± 1.5	56.0 ± 1.6	**2.17** ± 0.11	461.1 ± 1.5	**37.1** ± 1.5	4.60 ± 0.19
	30	493.1 ± 1.5	100.0 ± 2.2	**2.15** ± 0.13	492.5 ± 1.5	**92.4** ± 2.2	5.10 ± 0.31
Wdbc	0	0.6 ± 0.0	0.0 ± 0.0	0.51 ± 0.01	0.6 ± 0.0	0.0 ± 0.0	0.51 ± 0.01
	10	1447.3 ± 1.3	74.7 ± 2.0	**0.35** ± 0.01	1447.9 ± 1.3	**63.1** ± 2.0	3.74 ± 0.05
	20	1476.8 ± 0.9	137.6 ± 1.6	**0.41** ± 0.00	1477.6 ± 0.9	**132.5** ± 1.5	3.70 ± 0.04
	30	1488.6 ± 0.8	175.0 ± 1.6	**0.43** ± 0.00	1489.1 ± 0.8	173.5 ± 1.6	3.71 ± 0.04
Wine	0	0.1 ± 0.0	0.0 ± 0.0	0.49 ± 0.02	0.1 ± 0.0	0.0 ± 0.0	0.49 ± 0.02
	10	191.6 ± 1.3	28.2 ± 1.8	**0.23** ± 0.01	184.4 ± 1.0	**1.6** ± 0.5	0.96 ± 0.02
	20	215.7 ± 1.2	44.6 ± 1.5	**0.21** ± 0.01	210.5 ± 1.2	**14.0** ± 1.1	0.94 ± 0.04
	30	236.3 ± 1.1	65.9 ± 1.6	**0.19** ± 0.00	232.7 ± 1.1	**36.3** ± 1.5	0.89 ± 0.03



Iris Wine

Figure 5: Objective function value of SECM ($\mathcal{L}_s$ set, $\gamma = 0.5$ and $r = 1$) varying with the iteration number. X-axis uses a logarithmic scale.

4.7. Algorithms comparison

The $SECM_H$ and $SECMpl_H$ algorithms using $\gamma = 0.5$ are compared to $SFCM$ and $SKMEANS$ algorithms. The $SKMEANS$ method [31] is the semi-supervised k-means variant that handles crisp labels as prior information and generates a hard partition. The $SFCM$ [20] is the fuzzy variant of $SKMEANS$: it allows a probability on the labels and returns a fuzzy partition.

First, the $\mathcal{L}_s$ set is used. For $SFCM$, we set a probability to 1 for all constrained objects. The average ARIs varying with the percentage of constraints for all data sets are presented in Figure 6.

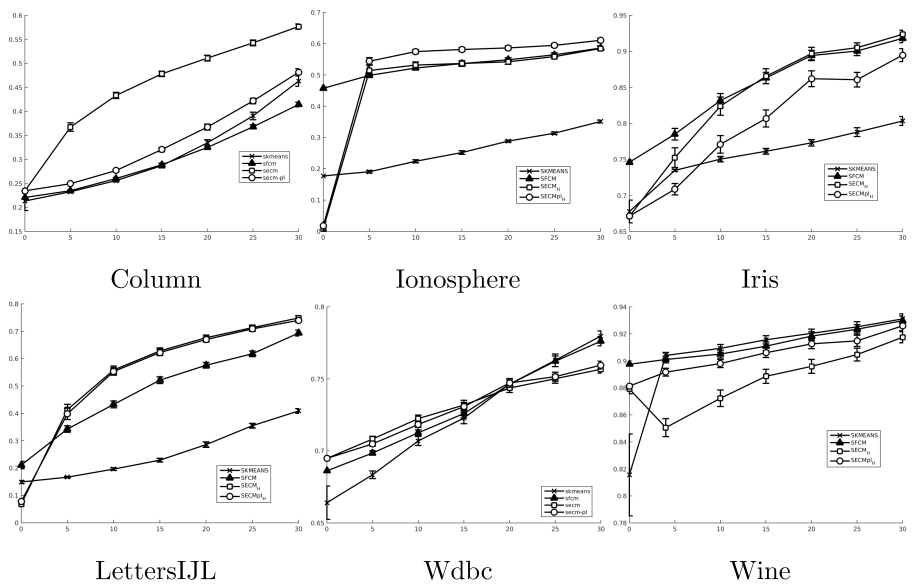


Figure 6: Algorithms comparison on the $\mathcal{L}_s$ set. The average of the ARI and the confidence interval is given on 100 trials.

The performances show that integrating label constraints leads to better define the clustering partition desired whatever the semi-supervised clustering algorithm. However, a decrease in performance can appear, as it can be seen for $SECMpl_H$ on Wine. This result is known to possibly occur when constrained objects are only in an overlapped area between classes [42].

As it can be seen, the $SECM$ algorithms bring good results on the data sets compared to the other algorithms except for the Wine data set.

A similar experiment is performed on the $\mathcal{L}_d$ set to compare semi-supervised clustering methods. The $SKMEANS$ algorithm is excluded since it can only take crisp labels. For the $SFCM$ algorithm, the imprecisions included in the $\mathcal{L}_d$ set and defined for an object by a subset of two labels are transformed into an uncertainty between the couple of labels by setting the probabilities of both labels to 0.5. Figure 7 presents the average ARI with respect to the percentage of constraints for all data sets.

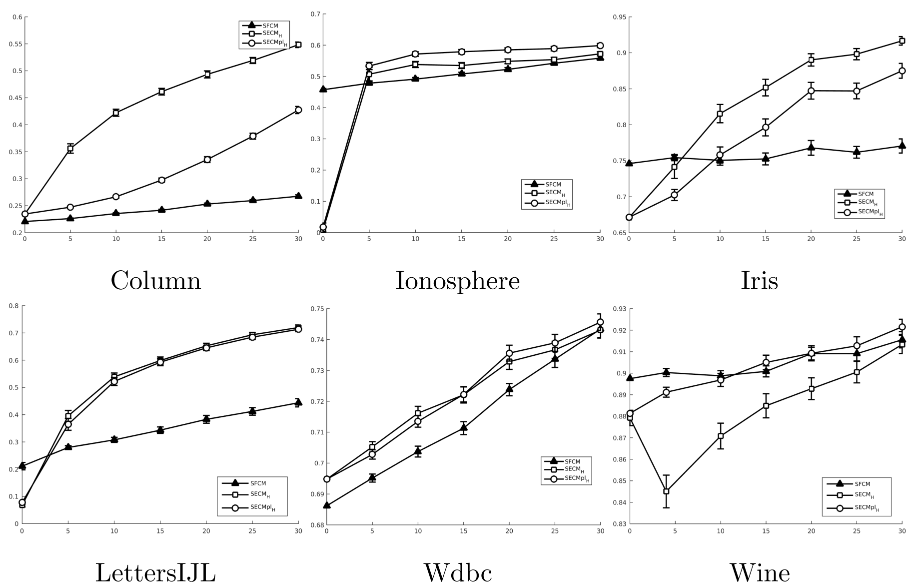


Figure 7: Algorithms comparison on the $\mathcal{L}_d$ set.

The results show that $SECM$ outperforms most of the time $SFCM$. This good performance for evidential semi-supervised clustering is explained by its ability to handle the imprecision coming from the background knowledge.

5. Conclusion

The SECM algorithm is a semi-supervised clustering variant of ECM based on the theory of belief functions. As such, it is capable of handling uncertainty and imprecision of the background knowledge and provides a credal partition that allows expressing imprecision and uncertainty regarding the assignment of objects to clusters. The SECM method, which is based on the minimization of an objective function, introduces prior knowledge in the form of label constraints. The optimization problem is solved using a Gauss-Seidel type method, with an alternate update of the masses and the cluster centroids. In this paper, a relaxation of constraints is proposed to fasten the optimization. The new heuristic, referred to as SECM-h, differs from the classical SECM as it provides a direct update formula for the masses. Experimental tests show that SECM-h achieves the same performance as its exact solution counterpart at a much lower execution time cost. Further investigations have also been conducted on the parameters of $SECM_H$ and $SECM_E$ and a comparison with other semi-supervised clustering algorithms has been performed. Results show good performance of the $SECM$ algorithms compared to other algorithms. Future work includes extending the ideas exposed in this paper to evidential semi-supervised clustering methods that handle pairwise constraints [24] or both labels and pairwise constraints [27].

References

[1] C.-W. Tsai, Seira: An effective algorithm for iot resource allocation problem, Computer Communications 119 (2018) 156–166.

[2] S. Sun, S. Wang, G. Zhang, J. Zheng, A decomposition-clustering-ensemble learning approach for solar radiation forecasting, Solar Energy 163 (2018) 189–199.

[3] A. Bilge, H. Polat, A scalable privacy-preserving recommendation scheme via bisecting k-means clustering, Information Processing & Management 49 (4) (2013) 912–927.

[4] S. Zahra, M. A. Ghazanfar, A. Khalid, M. A. Azam, U. Naeem, A. Prugel−Bennett, Novel centroid selection approaches for kmeans-

clustering based recommender systems, Information sciences 320 (1) (2015) 156–189.

[5] Y. Li, J. Sun, 3d magnetization inversion using fuzzy c-means clustering with application to geology differentiation, Geophysics 81 (5) (2016) J61–J78.

[6] M. Xue, L. Zhou, N. Kojima, L. S. dos Muchangos, T. Machimura, A. Tokai, Application of fuzzy c-means clustering to prtr chemicals uncovering their release and toxicity characteristics, Science of the Total Environment 622-623 (1) (2018) 861–868.

[7] J. Dongdong, N. Arunkumar, Z. Wenyu, L. Beibei, Z. Xinlei, Z. Guangjian, Semantic clustering fuzzy c means spectral model based comparative analysis of cardiac color ultrasound and electrocardiogram in patients with left ventricular heart failure and cardiomyopathy, Future Generation Computer Systems 92 (2019) 324–328.

[8] A. M. Anter, A. E. Hassenian, D. Oliva, An improved fast fuzzy c-means using crow search optimization algorithm for crop identification in agricultural, Expert Systems with Applications 118 (2019) 340–354.

[9] M.-H. Masson, T. Denœux, ECM: An evidential version of the fuzzy c-means algorithm, Pattern Recognition 41 (4) (2008) 1384–1397.

[10] T. Denœux, S. Li, S. Sriboonchitta, Evaluating and comparing soft partitions: An approach based on dempster–shafer theory, IEEE Transactions on Fuzzy Systems 26 (3) (2017) 1231–1244.

[11] C. Lian, S. Ruan, T. Denœux, H. Li, P. Vera, Spatial evidential clustering with adaptive distance metric for tumor segmentation in fdg-pet images, IEEE Transactions on Biomedical Engineering 65 (1) (2017) 21–30.

[12] S. B. Ayed, Z. Elouedi, E. Lefevre, ECTD: evidential clustering and case types detection for case base maintenance, in: IEEE/ACS 14th Inter-

national Conference on Computer Systems and Applications (AICCSA), IEEE, Hammamet, Tunisia, 2017, pp. 1462–1469.

[13] R. Abdelkhalek, I. Boukhris, Z. Elouedi, An evidential clustering for collaborative filtering based on users' preferences, in: International Conference on Modeling Decisions for Artificial Intelligence, Springer, Milan, Italy, 2019, pp. 224–235.

[14] M.-H. Masson, T. Denœux, RECM: Relational evidential c-means algorithm, Pattern Recognition Letters 30 (11) (2009) 1015–1026.

[15] K. Zhou, A. Martin, Q. Pan, Z.-G. Liu, ECMdd: Evidential c-medoids clustering with multiple prototypes, Pattern Recognition 60 (2016) 239–257.

[16] Z. Lui, Q. Pan, J. Dezert, G. Mercier, Credal c-means clustering method based on belief functions, Knowledge-Based Systems 74 (2015) 119–132.

[17] Z. Zhang, Z. Liu, A. Martin, Z. Liu, K. Zhou, Dynamic evidential clustering algorithm, Knowledge-Based Systems 213 (2021) 106643.

[18] T. Denœux, M.-H. Masson, EVCLUS: evidential clustering of proximity data, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 34 (1) (2004) 95–109.

[19] T. Denœux, S. Sriboonchitta, O. Kanjanatarakul, Evidential clustering of large dissimilarity data, Knowledge-Based Systems 106 (2016) 179–195.

[20] W. Pedrickz, J. Waletzky, Fuzzy clustering with partial supervision, IEEE Transactions on Systems, Man and Cybernetics 27 (5) (1997) 787–795.

[21] N. Grira, M. Crucianu, N. Boujemaa, Semi-supervised fuzzy clustering with pairwise-constrained competitive agglomeration, in: IEEE International Conference on Fuzzy Systems (FUZZ 2005), IEEE, Reno, U.S.A., 2005, pp. 867–872.

[22] S. Basu, M. Bilenko, A. Banerjee, R. Mooney, Probabilistic semi-supervised clustering with constraints, in: O. Chapelle, B. Scholkopf, A. Zien (Eds.), Semi-supervised learning, Cambridge, MA. MIT Press, 2006, Ch. 5, pp. 71–98.

[23] V. Antoine, N. Labroche, Semi-supervised fuzzy c-means variants: a study on noisy label supervision, in: International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Springer, Cádiz, Spain, 2018, pp. 51–62.

[24] V. Antoine, B. Quost, M.-H. Masson, T. Denœux, CECM: Constrained evidential c-means algorithm, Computational Statistics & Data Analysis 56 (4) (2012) 894–914.

[25] V. Antoine, B. Quost, M.-H. Masson, T. Denœux, CEVCLUS: evidential clustering with instance-level constraints for relational data, Soft Computing 18 (2014) 1321–1335.

[26] F. Li, S. Li, T. Denoeux, k-CEVCLUS: constrained evidential clustering of large dissimilarity data, Knowledge-Based Systems 142 (2018) 29–44.

[27] J. Xie, V. Antoine, On a new evidential c-means algorithm with instance-level constraints, in: N. Ben Amor, B. Quost, M. Theobald (Eds.), International Conference on Scalable Uncertainty Management. Lecture Notes in Computer Science vol 11940, Springer, 2019, pp. 66–78.

[28] T. Denoeux, NN-EVCLUS: neural network-based evidential clustering, CoRR abs/2009.12795. arXiv:2009.12795.

[29] V. Antoine, N. Labroche, V. V. Vu, Evidential seed-based semi-supervised clustering, in: International Symposium on Soft Computing & Intelligent Systems, Kitakyushu, Japan, 2014, pp. 706–711.

[30] V. Antoine, K. Gravouil, N. Labroche, On evidential clustering with partial supervision, in: International Conference on Belief Functions, Springer, Compiègne, France, 2018, pp. 14–21.

[31] S. Basu, A. Banerjee, R. Mooney, Semi-supervised clustering by seeding, in: Proceedings of 19th International Conference on Machine Learning (ICML), Sydney, Australia, 2002, pp. 19–26.

[32] G. Shafer, A Mathematical Theory of Evidence, Princeton Univ. Press. Princeton, NJ, 1976.

[33] P. Smets, R. Kennes, The transferable belief model, Artificial Intelligence 66 (1994) 191–234.

[34] P. Smets, The combination of evidence in the transferable belief model, IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (5) (1990) 447–458.

[35] J. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Advanced Applications in Pattern Recognition. Springer, Boston, MA, 1981.

[36] Y. Ye, An extension of Karmarkar's algorithm and the trust region method for quadratic programming, Springer, New York, NY, 1989, pp. 49–63.

[37] Y. Ye, On affine scaling algorithms for nonconvex quadratic programming, Mathematical Programming 56 (1-3) (1992) 285–300.

[38] D. Dua, C. Graff, UCI machine learning repository (2020).
URL `http://archive.ics.uci.edu/ml`

[39] L. Hubert, P. Arabie, Comparing partitions, Journal of Classification 2 (1) (1985) 193–218.

[40] C. Goutte, E. Gaussier, A probabilistic interpretation of precision, recall and f-score, with implication for evaluation, in: European conference on advances in information retrieval research, Springer, Granada, Spain, 2005, pp. 345–359.

[41] P. Smets, Imperfect information: Imprecision and uncertainty, in: A. Motro, P. Smets (Eds.), Uncertainty management in information systems, Springer, 1997, pp. 225–254.

[42] K. Wagstaff, S. Basu, I. Davidson, When is constrained clustering beneficial, and why?, in: 21st National Conference on Artificial Intelligence and 18th Innovative Applications of Artificial Intelligence Conference, AAAI Press, Boston, U.S.A., 2006, pp. 62–63.